UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/712,473 | 11/12/2003 | Robert E. Ober | J0658.0014 | 9335 |

38881 7590 07/08/2009

DICKSTEIN SHAPIRO LLP
1177 AVENUE OF THE AMERICAS 6TH AVENUE
NEW YORK, NY 10036-2714

| EXAMINER | |
|---|---|
| LI, AIMEE J | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2183 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 07/08/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/712,473 | OBER ET AL. | |
| | Examiner | Art Unit | |
| | AIMEE J. LI | 2183 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *20 April 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-22,25-34 and 37-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22,25-34 and 37-40* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 November 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

1. Claims 1-22, 25-34, and new claims 37-40 have been considered. New claims 37-40 have been added as per Applicants' request. Claims 1, 22, 25, and 29 have been amended as per Applicants' request.

### *Papers Submitted*

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as filed 20 April 2009.

### *Claim Rejections - 35 USC § 103*

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-5, 9-16, 20-22, 24-26, 29-32, 34-35, and 37-40 rejected under 35 U.S.C. 103(a) as being unpatentable over Hobbs et al., U.S. Patent Number 5,197,138 (herein referred to as Hobbs) in view of Wilson et al., U.S. Patent Number 6,944,736 (herein referred to as Wilson).

5. Regarding claim 1, Hobbs discloses a method for operating a multi-threaded system having a plurality of active threads (Hobbs col 1 lines 63-66), the method comprising:

a. assigning a global interrupt priority value to each of a plurality of interrupts ( Hobbs col 2 lines 46-50); and

b. processing a requested interrupt only when the interrupt priority value of the requested interrupt is higher than the interrupt threshold value (Hobbs col 2 lines 54-57).

6. Hobbs has not explicitly disclosed specifying a global interrupt threshold value that is applicable to all of the plurality of active threads. However, Hobbs has taught interrupts and traps occur in response to I/O requests (Hobbs column 2, line 62 to column 3, line 52). Wilson has explicitly taught specifying a global interrupt threshold value that is applicable to all of the plurality of active threads (Wilson Abstract; column 2, line 65 to column 3, line 55; and Figure 1, element 112). It would have been obvious to a person of ordinary skill in the art at the time the invention was made would have recognized that a latency manager as taught by Wilson minimizes unused processing time, thereby increasing efficiency of the system. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to increase system efficiency.

7. Regarding claim 2, Hobbs in view of Wilson discloses the method of claim 1, wherein processing the requested interrupt comprises: performing an interrupt entry process to prepare for an interrupt service routine (ISR); executing the ISR (Hobbs col 2 lines 6-13); and performing an interrupt exit process to return control from the ISR (Hobbs col 2 lines 14-19).

8. Regarding claim 3, Hobbs in view of Wilson discloses the method of claim 2, wherein performing the interrupt entry process comprises: identifying one of the plurality of active threads as an interrupt thread; switching to the interrupt thread if the interrupt thread is not executing (Hobbs col 2 line 67 to col 3 line 3); and branching to the ISR (Hobbs col 2 lines 6-13).

9. Regarding claim 4, Hobbs in view of Wilson discloses the method of claim 3, wherein each of the plurality of active threads comprises a thread context, and wherein performing the

interrupt entry process further comprises saving the thread context of the interrupt thread (Hobbs col 1 lines 57-64).

10. Regarding claim 5, Hobbs in view of Wilson discloses the method of claim 4, wherein performing the interrupt exit process comprises: executing a return from exception (RFE) instruction (Hobbs col 2 lines 14-19); and restoring the thread context of the interrupt thread (Hobbs col 1 lines 60-63).

11. Regarding claim 9, Hobbs in view of Wilson discloses the method of claim 3,

a. wherein each of the plurality of active threads consists of a first set of context registers (Hobbs col 1 lines 60-63) and a second set of context registers (Hobbs col 1 lines 60-63) *Note that the saving of the context registers is unique for each thread. The use of multiple threads (Hobbs col 1 lines 63-66) suggests a second set of context registers.*

b. wherein performing the interrupt entry process further comprises saving the first set of context registers of the interrupt thread (Hobbs col 1 lines 60-63), and wherein executing the ISR comprises: saving the second set of context registers of the interrupt thread if the second set of context registers of the interrupt thread are required for servicing the requested interrupt (Hobbs col 1 lines 60-63); servicing the requested interrupt (Hobbs col 2 lines 12-14); and restoring the second set of context registers of the interrupt thread after servicing the requested interrupt if the second set of context registers of the interrupt thread were required for servicing the requested interrupt (Hobbs col 2 lines 14-19).

12. Regarding claim 10, Hobbs in view of Wilson discloses the method of claim 9, wherein performing the interrupt exit process comprises: executing a return from exception (RFE) instruction (Hobbs col 2 lines 6-13); and restoring the upper context registers of the interrupt thread (Hobbs col 2 lines 14-19).

13. Regarding claim 11, Hobbs in view of Wilson discloses the method of claim 1, further comprising processing traps only in the active threads originating the traps (Hobbs col 2 lines 22-32). *Note that all the instances provided refer to responding to traps in the active threads that originate the trap.*

14. Regarding claim 12, Hobbs in view of Wilson discloses the method of claim 11, wherein processing traps comprises:

a. detecting a trap from an originating thread, the originating thread being one of the plurality of active threads (Hobbs col 3 lines 7-10);

b. storing trap background data for the trap if the trap is asynchronous (Hobbs col 3 lines 24-29); and

c. associating a trap pending indicator with the originating thread if the originating thread is not executing (Hobbs col 3 lines 40-50). *Note in line 44 the use of the term "delay". This suggests that there is an indicator waiting for a change to the originating thread.*

15. Regarding claim 13, Hobbs in view of Wilson discloses the method of claim 12, wherein processing traps further comprises:

a. performing a trap entry process to prepare for a trap handling routine (Hobbs col 3 lines 40-50);

b. executing the trap handling routine (Hobbs col 2 lines 12-14); and

c. performing a trap exit process to return control from the trap handling routine (Hobbs col 2 lines 14-19). *Note that, in Hobbs col 3 line 48, it is stated that an I/O interrupt routine is used for the trap, indicating that a trap uses the same methods for handling as typical interrupts.*

16. Regarding claim 14, Hobbs in view of Wilson discloses the method of claim 13, wherein each of the plurality of active threads comprises a thread context (Hobbs col 1 lines 63-66), and wherein performing the trap entry process comprises: waiting for originating thread to start executing if the originating thread is not executing (Hobbs col 2 line 67 to col 3 line 3); saving the thread context of the originating thread (Hobbs col 2 lines 6-13); and branching to a trap handler (Hobbs col 2 lines 12-14).

17. Regarding claim 15, Hobbs in view of Wilson discloses the method of claim 14, wherein waiting for the originating thread to start executing comprises monitoring the plurality of active threads until execution switches to one of the plurality of active threads associated with the trap pending indicator (Hobbs col 3 lines 40-50).

18. Regarding claim 16, Hobbs in view of Wilson discloses the method of claim 14, wherein performing the trap exit process comprises: executing a return from trap instruction; and restoring the context of the originating thread (Hobbs col 2 lines 14-19). *Note, again, that a trap is handled as an interrupt.*

19. Regarding claim 20, Hobbs in view of Wilson discloses the method of claim 13,

a. wherein each of the plurality of active threads consists of a first set of context registers (Hobbs col 1 lines 60-63) and a second set of context registers (Hobbs

col 1 lines 60-63), *Note that the saving of the context registers is unique for each thread. The use of multiple threads (Hobbs col 1 lines 63-66) suggests a second set of context registers*

b. wherein performing the trap entry process further comprises saving the first set of context registers of the originating thread (Hobbs col 2 lines 7-12), and wherein executing the trap handling routine comprises: saving the second set of context registers of the originating thread if the second set of context registers of the originating thread are required for servicing the trap (Hobbs col 3 lines 40-50); *Note that, as explained in the citation, a context switch may be necessary which, as indicated in Hobbs col 1 lines 57-63, requires saving the context into memory.*

c. servicing the trap (Hobbs col 2 lines 12-14); and restoring the second set of context registers of the originating thread after servicing the trap if the second set of context registers of the interrupt thread were required for servicing the trap (Hobbs col 2 lines 15-19).

20. Regarding claim 21, Hobbs in view of Wilson discloses the method of claim 20, wherein performing the interrupt exit process comprises: executing a return from trap instruction; and restoring the upper context registers of the originating thread (Hobbs col 2 lines 15-19).

21. Regarding claim 22, Hobbs discloses a method for operating a multi-threaded system having a plurality of active threads (Hobbs col 1 lines 63-66), the method comprising:

a. accepting a request for an interrupt;

b. switching execution to a predetermined one of the plurality of active threads (Hobbs col 2 line 67 to col 3 line 3); and

c. executing an interrupt service request from the predetermined one of the plurality of active threads to service the interrupt (Hobbs col 2 line 12-14),

d. wherein accepting a request for an interrupt comprises:

i. assigning a unique interrupt priority value to each interrupt for the multi-threaded embedded system (Hobbs col 2 lines 47-50); and

ii. taking the interrupt only when the unique interrupt priority value of the interrupt is higher than the global threshold interrupt value (Hobbs col 2 lines 54-57).

22. Hobbs has not explicitly disclosed specifying a global interrupt threshold value that is applicable to all of the plurality of active threads. However, Hobbs has taught interrupts and traps occur in response to I/O requests (Hobbs column 2, line 62 to column 3, line 52). Wilson has explicitly taught specifying a global interrupt threshold value that is applicable to all of the plurality of active threads (Wilson Abstract; column 2, line 65 to column 3, line 55; and Figure 1, element 112). It would have been obvious to a person of ordinary skill in the art at the time the invention was made would have recognized that a latency manager as taught by Wilson minimizes unused processing time, thereby increasing efficiency of the system. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to increase system efficiency.

23. Regarding claim 24, Hobbs in view of Wilson discloses a method for operating a multi-threaded embedded system having a plurality of active threads (Hobbs col 1 lines 63-66), the method comprising processing traps only in the active threads originating the traps (Hobbs col 2 lines 22-32).

24. Regarding claim 25, Hobbs in view of Wilson discloses a multi-threaded system comprising: thread execution logic for generating instruction requests from an executing thread (Hobbs col 2 lines 3-6); and threshold interrupt logic for generating an interrupt threshold value, wherein the thread execution logic only accepts interrupts having an interrupt priority higher than the interrupt threshold value (Hobbs col 2 lines 54-57).

25. Regarding claim 26, Hobbs in view of Wilson discloses the multi-threaded system of claim 25, further comprising interrupt thread logic for switching execution to a selected interrupt thread before servicing any interrupt (Hobbs col 2 lines 67 to col 3 line 3).

26. Regarding claim 29, Hobbs in view of Wilson discloses a multi-threaded system comprising: means for specifying a global interrupt threshold value; and means for processing a requested interrupt only when an interrupt priority value of the requested interrupt is higher than the global interrupt threshold value (Hobbs col 2 lines 54-57).

27. Regarding claim 30, Hobbs in view of Wilson discloses the multi-threaded system of claim 29, wherein the means for processing the requested interrupt comprises: means for identifying one of the plurality of active threads as an interrupt thread (Hobbs col 2 lines 3-6); means for switching to the interrupt thread if the interrupt thread is not executing (Hobbs col 2 line 67 to col 3 line 3); and means for branching to the ISR (Hobbs col 2 lines 12-15).

28. Regarding claim 31, Hobbs in view of Wilson discloses the multi-threaded system of claim 30, wherein the means for processing the requested interrupt further comprises means for saving a thread context of the interrupt thread (Hobbs col 1 lines 57 to 63).

29. Regarding claim 32, Hobbs in view of Wilson discloses the multi-threaded system of claim 31, wherein the means for processing the requested interrupt further comprises means for

restoring the thread context of the interrupt thread after executing a return from exception (RFE) instruction (Hobbs col 2 lines 14 to 19).

30. Regarding claim 34, Hobbs in view of Wilson discloses the multi-threaded system of claim 29, further comprising means for processing traps (Hobbs col 3 lines 7-11), the means for processing traps comprising:

a. means for detecting a trap from an originating thread (Hobbs col 3 lines 40-46);

b. means for storing trap background data for the trap if the trap is asynchronous (Hobbs col 3 lines 24-29);

c. means for processing the trap if the originating thread is executing (Hobbs col 3 lines 40-50); and

d. means for associating a trap pending indicator with the originating thread if the originating thread is not executing (Hobbs col 3 lines 40-50). *Note in line 44 the use of the term "delay". This suggests that there is an indicator waiting for a change to the originating thread.*

31. Regarding claim 35, Hobbs in view of Wilson discloses the multi-threaded system of claim 34, wherein the means for processing traps further comprises means for detecting execution of an active thread associated with the trap pending indicator (Hobbs col 3 lines 40-50). *Note that, clearly, for the AST to throw and interrupt when its originating thread is being executed, that state must be detected.*

32. Referring to claims 37-40, Hobbs in view of Wilson has taught wherein the global interrupt threshold value is a single global interrupt threshold value (Wilson Abstract; column 2, line 65 to column 3, line 55; and Figure 1, element 112).

33. Claims 6-8, 17-19, 27-28, 33 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hobbs et al., U.S. Patent Number 5,197,138 (herein referred to as Hobbs) in view of Wilson et al., U.S. Patent Number 6,944,736 (herein referred to as Wilson), as applied to claims 5, 16, 26, 32, and 35, and in further view of Radhakrishna, U.S. Patent Number 6,823,414 (herein referred to as Radhakrishna).

34. Regarding claim 6, Hobbs in view of Wilson discloses the method of claim 5, wherein performing the interrupt entry process further comprises thread switching (Wilson col 2 line 67 to col 3 line 3).

35. Hobbs fails to disclose disabling interrupts. Radhakrishna discloses an ISR that disables interrupts (Radhakrishna col 1 lines 66-67). In the environment of Hobbs, a system with the intention of organizing a large number and type of interrupts by giving them a priority system, one of ordinary skill in the art would be motivated to allow for an interrupt that is already in the ISR to maintain it's priority rather than potentially being "interrupted" by a further interrupt. As shown in Radhakrishna col 1 line 67 to col 2 line 4, interrupts are disabled "so that the currently-interrupting event can be processed in an unbroken fashion (i.e., without further interrupts from the same device that diverts CPU processing attention before the ISR has completed it's function)." It would have been obvious for one of ordinary skill in the art to allow the computer system of Hobbs to include the ability to disable interrupts while executing the ISR, as in Radhakrishna.

36. Regarding claim 7, Hobbs in view of Wilson and in further view of Radhakrishna discloses the method of claim 6, wherein performing the interrupt exit process further comprises

a. enabling interrupts (Radhakrishna col 2 lines 2-4) *Note that the interrupts are disabled so interrupts can not divert the "CPU processing attention before the ISR has completed its function", clearly suggesting that interrupts will be re-enabled after the ISR has completed.*

b. and thread switching (col 2 lines 8-19). *Note that the term "writes in memory contents to registers" suggests a context switch (or thread switch).*

37. Regarding claim 8, Hobbs in view of Wilson and in further view of Radhakrishna discloses the method of claim 6, wherein executing the ISR comprises enabling interrupts (Radhakrishna col 2 lines 2-4) and thread switching (Hobbs col 2 lines 8-19) after a predetermined interval (Hobbs col 2 lines 12-14). *Note that the complete ISR program code is considered to be the predetermined interval.*

38. Regarding claim 17, Hobbs in view of Wilson and in further view of Radhakrishna discloses the method of claim 16, wherein performing the trap entry process further comprises disabling interrupts (Radhakrishna103 with R from claim 6) and thread switching (Hobbs col 2 line 67 to col 3 line 3).

39. Regarding claim 18, Hobbs in view of Wilson and in further view of Radhakrishna discloses the method of claim 17, wherein performing the trap exit process further comprises

a. enabling interrupts (Radhakrishna col 2 lines 2-4) *Note that the interrupts are disabled so interrupts can not divert the "CPU processing attention before the ISR has completed its function", clearly suggesting that interrupts will be re-enabled after the ISR has completed.*

b. and thread switching (Hobbs col 2 lines 8-19). *Note that the term "writes in memory contents to registers" suggests a context switch (or thread switch).*

40. Regarding claim 19, Hobbs in view of Wilson and in further view of Radhakrishna discloses the method of claim 17, wherein executing the trap handling routine comprises enabling interrupts (Radhakrishna col 2 lines 2-4) and thread switching (Hobbs col 2 lines 8-19) after a predetermined interval (Hobbs col 2 lines 12-14). *Note that the complete ISR program code is considered to be the predetermined interval.*

41. Regarding claim 27, Hobbs in view of Wilson and in further view of Radhakrishna discloses the multi-threaded system of claim 26, further comprising disabling logic for disabling interrupts and thread switching while an interrupt is being serviced (Radhakrishna 103 with R from claim 6).

42. Regarding claim 28, Hobbs in view of Wilson and in further view of Radhakrishna discloses the multi-threaded system of claim 27, further comprising thread tagging logic for storing trap background data for asynchronous traps (Hobbs col 3 lines 7-10 and col 3 lines 24-29), wherein every trap is handled in its originating thread (Hobbs col 3 lines 40-50).

43. Regarding claim 33, Hobbs in view of Wilson and in further view of Radhakrishna discloses the multi-threaded system of claim 32, wherein the means for processing the requested interrupt further comprises means for disabling interrupts (Radhakrishna 103 with R from claim 6) and thread switching (Hobbs col 2 line 67 to col 3 line 3) during execution of the ISR (Hobbs col 2 lines 12-14).

*44.* Regarding claim 36, Hobbs in view of Wilson and in further view of Radhakrishna discloses the multi-threaded system of claim 35, wherein the means for processing the trap

comprises means for disabling interrupts (Radhakrishna 103 with R from claim 6) and thread switching (Hobbs col 2 line 66 to col 3 line 4 and col 3 lines 40-50). *Note that, as indicated in col 3 lines 15-20, the trap throws an interrupt, suggesting that it is treated as an interrupt.*

***Response to Arguments***

45. Applicant's arguments with respect to claims 1-22, 25-34, and 37-40 have been considered but are moot in view of the new ground(s) of rejection.

***Conclusion***

46. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Wilson et al., U.S. Patent Application Publications 2003/0005251 and 2003/0005252 are related publications to the patent cited above.

47. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

48. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

49. Any inquiry concerning this communication or earlier communications from the examiner should be directed to AIMEE J. LI whose telephone number is (571)272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

50. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

51. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Aimee J Li/
Primary Examiner, Art Unit 2183

5 July 2009